The Unofficial CMD SuperCPU FAQ

by Jeff Jones.

Q: What is a SuperCPU?

A: The SuperCPU is an accelerator, a plug-in attachment which makes your C-64 run faster.

Q: Why does it run faster?

A: It uses a faster 65816 processor or "brain" that runs at a higher clock speed than a normal C-64's brain, the 6502. It accomplishes this by using a combination of high-speed RAM, custom logic and a fast microprocessor that emulate the functions of similar components located inside your computer.

Q: How many times faster will my C-64 run?

A: Most programs will run 20 times faster. Optimized programs can run about 22 times faster.

Q: How does it work?

A: In layman's terms, it's really an external computer that runs 20 times faster. It has its own internal 64k memory.

Q: Can its internal memory be expanded?

A: Yes. The CPU can address 16 megabytes of RAM. CMD's One Megabyte RAMCard will be on the streets this spring.

Q: Can C-64 programs access the extended RAM like an REU?

A: Not like an REU because it's not an REU which is an external contraption. CMD's RAMCard will be seen like regular internal addressable memory. Native mode ML toolboxes or programs can easily access this memory with LDA, STA and all other forms of addressing.

Q: Do I need any extra hardware or software to run my SuperCPU?

A: Not if you're using it on a C-128 in 64 mode. Commodore 64 users may have the old flaky heavy brick power supplies. You need a heavy duty power supply. While the amount of power the SuperCPU uses is tiny, the common Commodore 64 power supply is known to be at best questionable, and was designed to just barely power a C-64 and maybe the smallest cartridge. The SuperCPU and computer together draw more power than a normal unit provides. It will likely burn out soon after you begin using the SuperCPU. If you have the power supply that came with your 1764 REU or if you're running on a C-128, you're fine.

Q: Can I use the SuperCPU on my SX-64 and can its power supply, which I can't replace easily, handle it?

A: The Sx-64's power supply has to run a monitor, amplifier, computer and disk drive. One would think the additional drain of a small SuperCPU would be negligible. Still this question has not been settled. Jim Brain reports that his SuperCPU works fine on his SX-64. Other Sx-64 units refuse to work at all with the SuperCPU.

Q: When will the SuperCPU for the 128 finally be released?

A: Officially the 128 unit took a back seat to the 64 CPU's RAM card. All I can say is probably later this year. CMD keeps re-discovering the Commodore computer as they design specialized hardware for it. There used to be an assumption that all Commodores were the same. CMD has discovered that many Commodores are physically different. Since timing is very important to proper operation, they have to fine tune the SuperCPU to run on just about any type of C-128.

Q: Can I use the SuperCPU with my existing cartridges?

A: With some. The SuperCPU has a pass-through port for cartridges, REUs or RAMLink. Most fastload cartridges will have to be disabled. Some ROM cartridges will work with the SuperCPU, including many game and some utility cartridges. When these are detected on power-up, the SuperCPU slows down to read from the cartridge's memory (the chips used in these cartridges are much too slow to keep up with the SuperCPU).

Cartridges that use the 'Ultimax' memory map won't work with the SuperCPU. This mode is selected when a cartridge pulls the GAME line low with the EXROM line set high on the cartridge port. Cartridges that use this mode include Action/Replay, Super Snapshot, Partner 64, most IEEE interface cartridges, the Lt. Kernal hard drive host adapter, and some game cartridges. Due to the way these cartridges replace the Kernal, they would render the SuperCPU inoperable. To use these cartridges, you'll most likely need to disable the SuperCPU.

Q: Does the SuperCPU come with any Software?

A: Yes. The unit comes with a small Q&D demo program that shows off the speed increase graphically. It also comes with an auto-executing GEOS program which you just copy to your boot disk. It allows you to boot GEOS without switching your SuperCPU to slow mode.

Q: Does the Unit come with JiffyDOS?

A: Of course. If you're planning on buying the SuperCPU, you won't need to buy an internal JiffyDOS for your computer.

Q: How compatible is it with my existing software?

A: That's a question that depends on your existing software. If you use one program over and over again, and that program doesn't work with the SuperCPU, it's not very compatible. If you have an eclectic collection of software, you'll find that most of the programs run well — and run much faster.

Q: Can I write software for the SuperCPU?

A: Yes. The SuperCPU is quite literally a C-64 clone. So if you can write C-64 software, you can write software for the SuperCPU. Just beware of illegal undocumented opcodes.

There is also the native mode of the processor. No runnable assembler exists for this mode yet, but programs written in the native mode will load and run on

a SuperCPU system, but take advantage of the chips richer language, registers, and memory addressing.

Q: You mentioned RAMLink. Is the SuperCPU compatible with RAMLink?

A: Yes. It was even designed with RAMLink in mind, and makes your RAMLink about 20 times faster. If you have a parallel HD cable for your RAMLink, it in turn, drives your CMD hard drive just that much faster.

There are some problems with older RAMLinks working with the SuperCPU. CMD will upgrade older RAMLinks for a nominal fee if you send them in. My own RAMLink had to be upgraded, while Fender's newer RAMLink works perfectly.

Q: Is it compatible with the CMD/PPI RAMDrive?

A: No. CMD reports that it would require rewriting the RD-DOS and incorporating that directly into the SuperCPU.

Q: What about the Quick Brown Box?

A: CMD and LOADSTAR haven't tested the QBB, but CMD Says it probably won't work.

Q: What about Serial port and User port devices?

A: According to CMD, all devices that attach to the User Port or Serial Bus should continue to be compatible when using the SuperCPU, provided they are used with compatible software. This includes such devices as Commodore-compatible modems, RS-232 interfaces and geoCable for the User Port, and disk drives, printers and parallel printer interfaces for the serial port.

Q: Why do a lot of my demos refuse to work with the SuperCPU?

A: Many demos use tricks that involve illegal opcodes, numbers that seem to yield predictable results with a 6502 series processor, but are meaningless to the SuperCPU, a whole other class of processor. Sometimes the demo might be fine, but might be packed with a packer that uses these illegal opcodes.

Note that this is why the programmer's reference guide always said not to use illegal opcodes — *Future expansion!*

Q: Will my REU slow down the SuperCPU?

A: No. These DMA devices are fully compatible with the SuperCPU. Programs do not need to slow the SuperCPU down to 1 MHz for REU access, as the SuperCPU itself automatically detects DMA transfers and controls the speed as needed. You will notice that during REU usage the Turbo light stays on.

Q: What about NovaTerm and SwiftLink?

A: Both NovaTerm and SwiftLink are flawlessly compatible. Other terminal programs seem to work flawlessly, however those that use software timing instead of hardware timing might run too fast.

Q: How do I optimize programs to run with the SuperCPU?

A: First note that optimization yields only a small increase in speed. The manual says that a simple FOR loop will operate more than twice as fast. This doesn't mean 40 times faster than a stock 64. It means 2 additional times faster than a stock C-64. You will go from 20 times faster to about 22 times faster. Going from 1 MHz to 2 MHz is a big step. From 20 to 22 is less noticeable, though I did notice that text screens scrolled much faster when optimized.

Q: What about Geos?

A: The Super CPU works with GEOS if you boot with the unit in the one megahertz mode, or if you install SuperGEOS, included on the disk that comes with the SuperCPU.
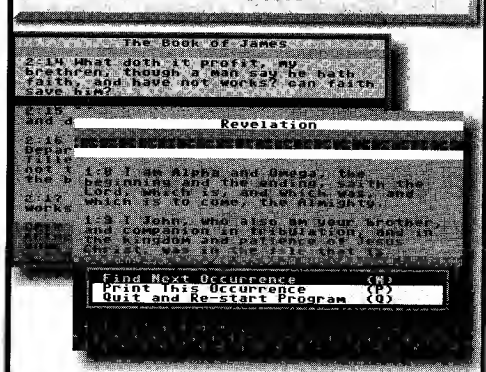
Q: How much is GEOS sped up?

A: Er, 20 times.

Q: Does that include GEOS disk access?

A: No. During disk access, speed drops to 1 MHz, but all processing during disk

IMPORTANT SUPERCPU MEMORY LOCATIONS

Location	Purpose
$D0741	(53364) GEOS Optimization (mirror VIC Bank 02, $8000-$BFFF)
$D0751	(53365) VIC Bank 01 Optimization (mirror $4000-$7FFF)
$D0761	(53366) BASIC Optimization (mirror $0400-$07FF)
$D0771	(53367) No Optimization (default; mirror all memory)
$D07A2	(53370) Software Speed Select - Normal (1 MHz)
$D07B3	(53371) Software Speed Select - Turbo (20 MHz) (*$D079)
$D07E2	(53374) Hardware Register Enable
$D07F2	(53375) Hardware Register Disable (*$D07D)
$D0B24	(53426) Bit 7: Hardware Register Enable Flag (1=Enabled)
Bit	6: System 1 MHz Flag (1=Enabled)
$D0B45	(53428) Bits 7 & 6: Optimization Mode Flags:
	00xxxxxx=GEOS Optimization Enabled
	01xxxxxx=VIC Bank 01 Optimization Enabled
	10xxxxxx=VIC Bank 02 Optimization Enabled
	11xxxxxx=No Optimization
$D0B56	(53429) Bit 7: JiffyDOS Switch Flag (1=Enabled)
Bit	6: Speed Switch Flag (1=1 MHz)
$D0B64	(53430) Bit 7: Processor Emulation Mode Flag (1=Emulation)
Bit	6: Reset Switch Flag (1=Switch Pressed)
$D0B84	(53432) Bit 7: Software 1 MHz Flag (1=1 MHz)
Bit	6: Master 1 MHz Flag (1=1 MHz via any source)
$D0BC5	(53433) Bit 7: DOS Extension Mode Flag (1=Enabled)
Bit	6: RAMLink Hardware Registers Flag (1=Enabled)
$D200-$D2FF4	(53760-54015) RAM
$D300-$D3FF5	(54016-54271) RAM (available for user programs)

Notes:
1. Write only, hardware registers must be enabled to activate location.
2. Write only, active with hardware registers enabled or disabled.
3. Write only, active with hardware registers enabled or disabled, but does not over-ride hardware Speed switch.
4. Read only with hardware registers disabled, Read/Write with hardware registers enabled, write access reserved for system only.
5. Read only with hardware registers disabled, Read/Write with hardware registers enabled.
6. Read only with hardware registers enabled or disabled (write with hardware registers enabled has no effect).
*   Duplicate register location.
IMPORTANT NOTE: Enabling the SuperCPU hardware registers also causes some changes in the Kernal ROM memory map ($E000-$FFFF). To avoid problems, do not leave the hardware registers enabled any longer than necessary.

activity is sped up. So while you won't access GEOS files 20 times faster, there will be a net gain.

Q: How fast do GEOS screens update.

A: Typically, you can't see them draw.

Q: What have you discovered at LOADSTAR that wasn't written in the manual?

A: Well, it's a good idea not to change your fonts directly after going into the optimized mode. Wait at least one Jiffy. On Fender's system this isn't needed, but without it, my system can display garbage.

Also don't use the CINT command and change banks while in the optimization mode. Optimize after all this is done to avoid incredible garbage on the screen.

Argument For Purchasing a CMD SuperCPU

By Jeffrey L. Jones. Many people are unsure about the SuperCPU. If you're a satisfied Commodore user, why would you want one? Well whether we want to admit it or not, we like speed. We also like hardware. Personally, I shiver at the thought of having to disengage my SuperCPU in order to go in to the 128 mode. Two megahertz never seemed that much faster, but twenty megahertz on a Commodore seems like 100 megahertz on my Windows PC.

Where will you be disappointed? Most likely in disk access speed. All disk access occurs at 1 MHz, no slower than you're used to, but slow once you become used to 20 MHz. Other than that, once a program boots, the program, whether it be a Database or word processor, runs 20 times faster.

When I use Microsoft Word or WordPerfect on my 40 MHz Amiga or 75 MHz PC, my text files never seem to become bogged down, no matter how long they become. Anyone who has ever used Speedscript or TWS on a long text

file knows that it can take a half second for each character to appear in a long text file. Now I enjoy how The Write Stuff can edit the largest text files with no delay. No matter how large the file, I can't type too fast for TWS to keep up. Oh, the delay is still there, but the half second delay is now twenty times shorter or twenty-five thousandths of a second. I can live with *that* kind of delay!

I know what some people are asking: Why scream? Well the whole point of computers is to do things faster than humans can do. Like sand in the hand, we defeat the jumps in computing speed by writing more ambitious programs. As computers grow faster, we begin to ask them to do more complex things, we bog them down more — slowing down computers. Frankly there were programs never written for the C-64 because even in machine language, they'd be too slow to be useful. Now I can see word processors that automatically fix errors as we type, and offer graphical expressions when before it would have crawled. With an REU or ram card plugged in, the SuperCPU can perform tasks like these while you type with

no humanly noticeable delay.

I can do all sorts of things now in seconds that I wouldn't think about doing before. Archiving files used to be the last thing I wanted to do. Now I do it all the time because on my RAMLink, all of my archivers (except the Compression Kit which won't work with the SuperCPU) operate much, much faster.

Most well-written games won't run so fast that you can't win because timing loops will keep the play paced. But any program that computes will compute much faster. Some BASIC games, so slow that you barely like them, are suddenly zippy and challenging. With games, compatibility is a crapshoot, but you'll likely find a

fantastic increase in the playability of your board games, especially chess and the like. If at the highest level it took your chess program nine minutes to move, imagine that same move in 27 seconds.

Take a nostalgic look back in your LOADSTAR library of disks and you'll re-discover hundreds of games and programs that suddenly work 2000% better.

I remember my RAMLink seemed to LOAD files instantly when I first got it. The novelty wore off after about a month and my RAMLink was reduced to the ranks of merely "very fast." Now the noticeable speed increase of the SuperCPU is four months old for me and it's still a pleasure.

I got this E-mail from Louis Gilgo:
I saw that you asked about the SuperCPU in comp.sys.cbm and wanted to add my two cents. I've used two of them for months now, and overall I'm quite pleased.

For most of the "serious" software I use, the SuperCPU works perfectly and speeds things right along. This includes a spreadsheet (Pocket Planner 2), a couple of word processors, a couple of compilers (C and Forth), and the Buddy resident assembler (EBUD) which is now usually finished before by finger has left the RETURN key.

*For most games the SuperCPU has no effect, but a few are greatly improved by its presence. Both Test Drive and Space Station Oblivion are **much** more enjoyable to play. They literally crawl without it.*

The one place I have had problems, however, was with Geos. I haven't narrowed down the actual culprit yet, so I haven't bothered to get in touch with CMD, but when I exit either geoPaint or geoWrite with the SuperCPU enabled, without exception the directory in which the document I was working on gets trashed. Somehow the track and sector pointers of the second directory block (track 18, sector 4 I think) get overwritten with zeros, which of course causes an "illegal track or sector" error any time you try to look at the directory. I can fix this manually with a sector editor, but it would be nice if the problem didn't occur in the first place.

Excluding the Geos problem, I'm amazed how compatible software is with the SuperCPU. CMD did a pretty decent job in my opinion.

BTW, keep up the good work on LOADSTAR and LOADSTAR Letter. I look forward to both each month. Cheers,

Louis Giglio
I've used the SuperCPU with GEOS with no problem, but not every Commodore system is the same. See the unofficial FAQ. But perhaps the strongest argument for buying the SuperCPU is the fact that Louis has purchased two.

The Mysterious SuperCPU Native Mode

by Robin Harbron. I've got a bit of a program for you here, that actually requires CMD's SuperCPU. It isn't astounding at all, it merely makes a somewhat fancy display, and reveals some fundamental techniques necessary for programming the SuperCPU. It also reveals a little of how the SuperCPU manages to get along with its host, the C64.

When your SuperCPU is turned on, it is set to something called *Emulation mode.* The 65816 CPU inside your SuperCPU has this mode built in, which emulates a 6502, which is similar to the 6510 inside a C64. This is what allows the SuperCPU to provide a high level of compatibility with your old 8-bit C64 programs. However, you can put the 65816 into its native mode, which allows you to use the full 16-bit power your SuperCPU is capable of.

In native mode you can select whether the accumulator (& associated memory operations) are 8-bit or 16-bit. You can also select the X and Y

registers to be 8 or 16-bit. X and Y are changed together, so either they are both 8-bit or both 16-bit. 16-bit registers are definitely something to be excited about!

In Figure 1 is a list, adapted from Doug Cotton's in Commodore World #18, of code to use to swap between all the different modes mentioned above:

Most people don't have access to a 65816 native mode assembler yet, so I've included the .byte values to enter into your assembler for the new opcodes. You'll see 3 new commands here:

xce ($fb) This command allows you access to the emulation bit of the status register. You simply set the carry for emulation mode, or clear it for native mode and issue the eXchange Carry and Emulation flags command.

rep ($c2) REset (or clear) Processor status bits. Any bit that is set in the operand is cleared in the Processor Status Register. Bit 4 controls the size of the index registers, and bit 5 controls the size of the accumulator, where 1 = 8-bit, and 0 = 16-bit.

sep ($e2) SEt Processor status bits. Any bit that is set in the operand is set in the Processor Status Register.

Now that you can get in and out of all the various 65816 modes, I'll explain a few parts of the program we're working towards in Figure 2.

Note that we're switching to 8-bit mode before waiting for raster line 50. If we didn't, we'd have to load a 16 bit number into the accumulator to use in the CMP $D012 line. More on this later. Also, the compare would be done between the 16-bit accumulator and not just $D012, but on $D012 and $D013! And this is not at all what is desired here.

Macro Name	6510 ASSEMBLER	65816 ASSEMBLER	MODE
gonatv	clc	clc	Native
	.byte $fb	xce	
go16	.byte $c2,$30	rep #$30	All 16 bit
go16xy	.byte $c2,$10	rep #$10	16 bit X&Y
	.byte $e2,$20	sep #$20	8 bit A
go16a	.byte $c2,$20	rep #$20	16 bit A
	.byte $e2,$10	sep #$10	8 bit X&Y
go8	.byte $e2,$10	sep #$30	All 8 bit
goem	sec	sec	Emulation
	.byte $fb	xce	

Figure 1

```
NATIVE MODE PROGRAM LISTING
*= $4000   ;assemble to 16384
sta 53374  ;put SUPERCPU in basic
sta 53366  ;optimization mode
sta 53375  ;(screen at $0400)
lda #127   ;disable CIA #1
sta 56333  ;interrupts
sei
lda #147   ;clear the screen
jsr $ffd2
lda #0          ;background & border
sta $D020       ;to black
sta $D021
clc             ; (gonatv)
.byte $fb       ;xce
.byte $e2,$30;sep #$30 (go8)
start   lda #50 ;wait until raster
line1      cmp $d012 ;line 50
bne line1
.byte $c2,$10;rep #$10 (go16xy)
.byte $e2,$20;sep #$20
lda #0
ldx #0     ;16 bit ldx #$0c00
.byte $0c
loop1     .byte $1a      ;inc a
sta $d021
.byte $1a       ;inc a
sta $d021
.byte $1a       ;inc a
sta $d021
dex             ;16 bit
bne loop1
.byte $e2,$30;sep #$30 (go8)
jsr $ff9f ;check for keypress
jsr $ffe4
beq start       ;loop if none
sec             ; (goem)
.byte $fb       ;xce
sta 53374       ;turn off
sta 53367       ;optimization
sta 53375
lda #129        ;enable CIA #1's
sta 56333       ;interrupts
cli
rts
               Figure 2
```

We then put the index registers into 16-bit mode, and I load the X register with a 16-bit value. Since my assembler doesn't understand doing a 16-bit immediate load, I have to cheat a bit:

```
          ldx #0
          .byte $0C
```

loads the value $0C00 into the 16-bit X register. This works because the 65816 uses the same opcode for LDX whether X is 8 or 16-bit. If it's in 8-bit mode, it expects the next byte to be the operand; in 16-bit mode, it expects the next 2 bytes in standard low-byte, high-byte format.

With these 16-bit registers, you can see how easy it is to make a much bigger loop than with the 6510. You can address the entire C64's memory with a single simple loop!

The final new thing in this program is the use of the INC A command, which shows as byte $1a in the program, as my assembler doesn't understand that command. I can't count the times I wish this command existed on the 6510. In fact, pretty much all the commands you ever wished for are available on the 65816. There is a great list in Commodore World #16, check it out!

Now, when you run the type-in program in figure 3, you will be treated to a (fairly) attractive display, which is a series of diagonally running colored stripes. These are created merely by changing the background color very rapidly. There are a couple of things to study on this display which will give us some insight into how the SuperCPU works.

Some of the stripes are 8 pixels wide, while others are 16 pixels wide. A new stripe is made every time one of the STA $D021 commands is executed. The 8 pixel wide stripes are created by the second and third STA $D021's that are only separated by the INC A's, while the 16 pixel wide stripes are created by the first STA $D021 that has the extra overhead of the DEX and BNE commands. By the time the STA gets executed, the horizontally-sweeping raster has already moved along further to the right. If it had even more to do, the stripes could be 24 pixels wide, or more.

Why multiples of 8 pixels? This ties in to the original 1 MHz C64, which the SuperCPU is still somewhat bound to. Every time the SuperCPU writes to memory (e.g. a STA command), the memory has to be transferred through to the C64's memory, so the VIC can have the data available to display it. The VIC cannot see the SuperCPU's memory, only the original memory in the C64. Now, since the SuperCPU is running 20 times faster, it has to slow down to the C64's 1 MHz speed to write the byte. So if the SuperCPU's program is doing a lot of writing to memory, it will work at a much slower effective speed than 20 MHz. However, the optimization modes reduce the area of memory that gets "mirrored" this way, allowing the SuperCPU to run closer to its potential. This is why if you are in an inappropriate optimization mode, the computer will have seemed to have locked up. It hasn't, the VIC just can't see the changes being made to the screen, so you don't see them either.

However, when you write to a hardware register (such as the background color, $D021), it can only do 1 write in each 1 MHz C64 cycle. So when we try to do a bunch of successive writes to $D021, the computer waits until the previous write was successful to do the next. Now, the raster moves across the screen at 8 pixels per 1 MHz cycle, so you can see why the stripes are made of 8 pixel multiples.

Also on the screen, you'll see long horizontal lines every 8 pixel lines. This too has to do with the VIC. Normally in the C64, the VIC and the 6510 takes turns sharing the 64's memory. Every cycle the VIC gets access to the memory for the first

half, then the 6510 for the second half. But every 8 raster lines, the VIC needs to grab more information to keep up the display than it can get. So it "stuns" the 6510, and uses the 6510s turns as well as its own to get the data it needs, and when it's done, allows the 6510 to start chugging away again.

In many circles this is called a "bad line", as it slows the 6510 down. You can stop these bad lines by turning off the VIC's display (bit 4 of $D011 controls this), but of course for most uses we need the display. Incidentally, controlling these "bad lines" is the basis for many of the cool VIC effects that demo coders are capable of.

Those long horizontal lines are the result of these "bad lines". It happens like this: The background color is changed to light red (pink), and suddenly the VIC takes over the 6510's memory access turns. The SuperCPU, oblivious to this, wants to write the next color to $D021 (a dark gray). It waits until memory becomes available to write to, and has to wait, and wait, and wait, as it needs use of the 6510's memory access turns, which aren't available. Eventually the VIC allows control back to the 6510, and the SuperCPU writes the byte through. But the result on the screen is this long line.

There are many more things to learn about the SuperCPU, and I am just a beginning student. But armed with enthusiasm and some diligence, we can all learn to make more use of these powerful units. Let me know what you'd like to see in the way of SuperCPU specific articles!
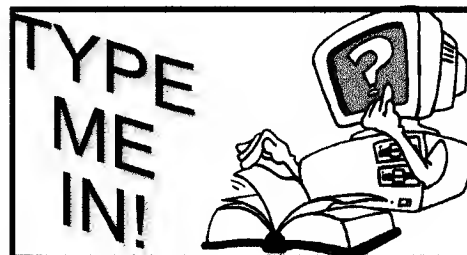

```
10  x=16384
20  reada:ifa=-1then40
30  b=b+a:pokex,a:x=x+1:goto20
40  ifb<>12569thenprint"data error":end
50  sys16384
100 data 141,126,208,141,118,208,141
110 data 127,208,169,127,141, 13,220
120 data 120,169,147,32,210,255,169
130 data   0,141, 32,208,141, 33,208
140 data  24,251,226, 48,169, 50,205
150 data  18,208,208,251,194, 16,226
160 data  32,169,  0,162,  0, 12, 26
170 data 141, 33,208, 26,141, 33,208
180 data  26,141, 33,208,202,208,241
190 data 226, 48, 32,159,255, 32,228
200 data 255,240,215, 56,251,141,126
210 data 208,141,119,208,141,127,208
220 data 169,129,141, 13,220, 88, 96
230 data -1
```

Note, this program was intended to be RUN on a CMD SuperCPU system.

Figure 3

One Stop Commodore Surfing

by Sherry Freedline. There's a new site on the horizon that's about to revolutionize the Internet's Commodore Community! By this time next year, this all new site will be as well-known among Commodore users as AOL is among PC users. The man behind the site, is none other than the creator and owner of the infamous U.S. Commodore 8-bit Computer WWW Site; Jim Brain.

Jim's original Commodore site was developed in 1994 while learning the ins and outs of HTML. As Jim's original site stands today, it's won five awards and has been accepted as a member of the Commodore Ring.

Currently his site offers the Internet Community access to the following features: Commodore Hacking, CaBooM - Jim's Commodore related Links Search Engine, the Commodore FAQ, documentation on the Commodore's internals, Jim Brain's infamous Commodore Trivia, Craig Bruce's or Marko Makela's Picture Pages, the Canonical List of Commodore Produced Computer Equipment, and CBM Magazine Index Links. Wow! What more could the average Commodore user want? Believe it or not, Jim's decided that we do indeed need and/or want more features.

Since 1994, Jim has learned more about our needs and our common uses of the Internet. And, naturally, over the years, Jim has increased his knowledge of HTML and other Internet programming languages. So, in August of 1996, he began developing a site to accommodate these needs. Once unveiled, his new site will be introduced to the world as the "Virtual Internet Commodore User's Group", or more simply, "VICUG".

According to Jim, the prototype was completed in September of 1996, and opened for limited beta usage in February of 1997. If all goes well, VICUG will be open to the public by the end of April.

According to Jim, VICUG will be a "virtual user group, complete with SIGS (Special Interest Groups), a file library, a virtual BBS, a reference library, a joining application (to become a member of virtual user group), along with some goodies you won't find anywhere else, and will have a question and answer area." (Hmmm... I can't wait to investigate the secret goodies for myself!) Jim has also kindly revealed that he is working on an "automated Press Release System and that he and Dale Sidebottom are discussing the creation of a Newsletter Editor Helper System that

would allow User Group Newsletter editors to post and retrieve NEW and never before published articles for their publications." Wow, aren't these thoughts amazing? And if anybody can bring them to life, I'm sure Jim can!

The above snippet of information reminded me of all the comforts I have come to expect from today's on-line services such as Genie, Delphi, and etc. This is exactly what the Internet needs! Personally, I feel that such a similarity will help draw more Commodore users into the Internet environment, basically because they will feel at home and yet easily be able to learn about the new features only available via the Internet Commodore environment.

Some of VICUG's goodies are already in place. In fact you may have already accessed them without knowing they were a part of Jim's new site, CaBooM (http://www.jbrain.com/CaBooM/) was the first VICUG "goodie" to be unveiled. I'm sure many of you have already used it to find a Commodore Link or have even added the CaBooM search engine HTML code to your own personal web site. Another new feature is "C= and Swap: the Commodore Classifieds", which has been needed for quite some time. I'm sure many of you get tired of dodging all the For Sale postings all too frequently posted in the comp.sys.cbm newsgroup. Now, when you want to post something for sale, or if you are looking to buy, there's finally a place on the 'net specifically for that purpose. C= and Swap can be found at http://www.jbrain.com/cswap/. Commodore Hacking: (http://www.jbrain.com/chacking/) and Commodore trivia: (http://www.jbrain.com/trivia/) have also been moved to Jim's new server.

Think back to those days when you were first learning how to connect your computer to the on-line world. How often did you wish for your own personal tutor that could come to your house and take you by the hand and show you all the ins and outs of the on-line world? Well, now, Jim has found a way to virtually supply that helping hand. Now, all one needs to do on their own is connect to Internet and locate VICUG. From there, the rest is all downhill! Just when I thought the Commodore Internet World couldn't get any better, Jim proves me wrong. Thanks Jim for keeping our "world" alive and kicking!!

Sherry Freedline is the proud owner of QT's Dream Space!
http://www.lm.com/~qt
Genie's Commodore 64/128 RT Sysop
qt@telerama.lm.com
qt@genie.com

Taking Over Your Machine

by Robin Harbron. I like reading a good mystery, but I really don't like to have mysterious stuff going on inside my C64. Even in a relatively simple computer, many activities go on behind the scenes. In the 64, from the time you turn on the machine, an IRQ (Interrupt ReQuest) is programmed to execute 60 times a second. Now this is a blessing for the programmer who wants the computer to handle mundane tasks such as checking the keyboard, updating the system clock (TI & TI$), and so on. All this is done transparently, especially to the BASIC programmer, through this interrupt. The computer simply stops executing your program, briefly does its housekeeping, and then continues with your task.

However, when you are doing timing sensitive tasks, especially the sort used in games & demos, these interrupts can cause glitches to appear on the screen. For example: Your program is to change the display in some way, say, change the background color, and then change it again. Perhaps it would be a screen with a black background, and then a horizontal white bar for a portion of the screen, then black again. The problem is this: If this housekeeping IRQ occurs just before the background is changed, the computer will be busy "servicing" the IRQ, rather than watching for your effect. The end result is that the display flickers. Before I understood this, I was frustrated to no end, wondering what mysterious things my computer was doing. I wanted control!

The solution to this is a simple bit of machine code. In fact, what I present here is a bit of overkill, but it's good to be using this code already, for when we discuss programming raster interrupts.

If you RUN the program in Figure 4, you'll see the problem: the screen will flicker from time to time. The housekeeping IRQ is keeping the computer busy and causes it to miss the color change from time to time. Now, remove the REM's in lines 50 and 110. These lines contain the enable and disable IRQ code. Now run the program. The flashing stops! You may notice each time you run the program, you get a slightly different display on the screen. Sometimes the borders between the black and white are a straight line, sometimes a little jagged. If this program were more complicated, you may still see flicker left to right along those lines (from 8 to 56 pixels worth). But this is getting far beyond the scope of this article,

```
        *= $4000 ;assemble at 16384
              lda #127 ;disable CIA #1 interrupts
              sta 56333
              sei

start         lda #100 ;which raster line to work on
line1         cmp $d012;is it there yet?
              bne line1;loop & compare until it is
              lda #1   ;change background & border to white
              sta $d020
              sta $d021

              lda #120 ;which raster line to work on
line2         cmp $d012;is it there yet?
              bne line2;loop & compare until it is
              lda #0   ;change background & border to black
              sta $d020
              sta $d021

              jsr $ff9f;use kernal routines to scan keyboard
              jsr $ffe4
              beq start;and loop if there is no keypress

              lda #129 ;re-enable CIA #1 interrupts
              sta 56333
              cli
              rts
```

```
To disable all IRQs: lda #127 ;stop CIA #1 from generating IRQs
                     sta 56333;same as POKE 56333,127
                     sei      ;tell 6510 to ignore all IRQs

To enable IRQs:      lda #129     ;start CIA #1 generating IRQs
                     sta 56333    ;same as POKE 56333,129
                     cli          ;tell 6510 to act on IRQs
```

although I plan to explore these issues later.

This program uses a technique called polling to split the screen. This technique is often very inefficient, but it is easy to program and understand. You should also note that this program runs much better with the SuperCPU on, even without the IRQs disabled, as the extra speed makes the housework get done so much faster.

Here's the source code for the little example, given in Turbo Assembler format:

Again, this isn't at all an efficient way of programming a raster routine, but I'm convinced it's the easiest to implement and understand. It's the way I first learned, and then I went on later to understand interrupts. I will explain those later.

All it takes to understand this routine is the following: Your C64 & monitor are in sync. 60 times a second the monitor redraws your display, from top to bottom,

and your computer feeds it the information it needs, as it goes. Your screen is made up of 262 (312 outside of North America) horizontal lines, stacked one on top of another, numbered 0 - 261. Lines 50-249 make up the display between the top and bottom borders. A counter located at $D012, counts along with the current line being drawn. Since we need 9 bits to count to 261, the high bit of $D011 is used as the 9th bit of $D012. $D012 counts from 0-255, then bit 7 of $D011 turns from a 0 to a 1. $D012 then counts from 0-5. Then bit 7 of $D011 goes back to a 0, and $D012 starts counting over again from 0.

```
HERE'S A SHORT PROGRAM THAT ILLUSTRATES THE
                 PROBLEM AND SOLUTION:

10 X=16384
20 READ A : IF A=-1 THEN GOTO 40
30 POKE X,A : X=X+1 : GOTO 20
40 SYS 16384
50 REM DATA 169,127,141,13,220,120
60 DATA 169,100,205,18,208,208,251,169
70 DATA 1,141,32,208,141,33,208,169,120
80 DATA 205,18,208,208,251,169,0,141,32
90 DATA 208,141,33,208,32,159,255,32
100 DATA 228,255,240,218
110 REM DATA 169,129,141,13,220,88
120 DATA 96,-1
```

Figure 4

All the program does is wait for the raster to reach a certain point (the waiting is what makes it inefficient) and then changes the background color, then changes it again, 60 times a second.

This is an excellent program to experiment with. The best way to learn is to take a little program like this, and change it into something that is your own.

I welcome your comments & suggestions. E-mail me at macbeth@tbaytel.net.

False Color

By John Elliot. Thanks to Hugh McMenamin's dithering approach, we can print in color if the source image is in color. The magenta, yellow and cyan components of the original picture are saved as separate files. When they are sequentially printed, with a back up of the paper after each printing, the overprints merge into all the color of the rainbow. Hugh's LOADSTAR program works with a monochrome printer so long as the correct single color ribbon is used for each print. His Illustrator for The Write Stuff word processor, can place color pictures within a page of text. Either a color or monochrome printer can be used.

I wanted to convert my black and white pictures to color. Hugh sent me a monochrome picture from LOADSTAR that he had loaded into Runpaint and colored by hand. He told me that "It was hard work, but worth the effort." I have scanned or digitized pictures from coloring books and colored them in a paint program. Most of the shapes are large and closed, so that the "fill" command will give solid blocks of different colors.

Most of my monochrome pictures though, are scans created by Ben Johnson's 4-scan process. Ben's $70 program does not give a high definition copy of single line drawings.

It is very good though at showing gray scale ranges. Portraits come out especially well, because of 4-scan's ability to copy ranges of skin tone and shadow. Although there is a program which will convert 4-scan images to doodles, I have trouble hand coloring these pictures which do not use solid blocks of color.

I asked Ben if he had considered creating a color scanner for Commodore users. He said that it would be possible using successive scans using magenta, yellow and cyan filters, and then to print as overlays the resulting files.

A simpler alternative that he and I have both tried is the use of "false color". The Hubbell telescope uses this approach. The images it captures on film are shades of gray rather than color. Specific colors are assigned in the print outs to each gray tone.

The magazine industry uses false color so frequently and effectively that it is difficult to know when natural color is used. The process can be used to change one color for another, or make a black and white picture multicolored.

The 4-scan scanning program allows the user to adjust the exposure (darkness) of the scanned image before it is saved to disk. Each level of darkness emphasizes different aspects of the object or face. I successively saved a scan as high exposure (light image), normal, and low exposure. I then printed, with back up, each of the files sequentially using a yellow, magenta, and cyan ribbon. I ended up with a multi-colored picture of Joe Stalin, even though the original was in black and white.

I do not want to exaggerate the quality of the picture. 4-scan creates images that are closer to an Impressionist painting than to a snapshot. The false color pictures share this limitation.

That this program allows a wide range of exposure saves means that the Commodore has the freedom of a darkroom.

If the color picture emphasizes the wrong features, color control is possible by resaving the original image at different exposures.

Encouraged by my 4-scan results, I tried something similar with my Final Cartridge III. My Final Cartridge III can freeze and do a screen dump to the printer in color. If the screen image is in color, the results on paper duplicate the screen. I have control over height to width ratio, size, and intensity of the printed image. The FINAL CARTRIDGE III also permits me to, after freezing the screen, cycle through a range of colors for the foreground and background of the picture, as well as for the border.

4-scan prints using a blank screen, in order to speed up the process. Ben has made however, a program which will convert 4-scan images to doodles. I used this program to convert Joe Stalin to a doodle. Using doodle (Any hires program would do-Runpaint by just changing the prefix of the file name.), I showed the monochrome image on my screen. I froze it with FINAL CARTRIDGE III, cycled through my color options, and printed Joe with a yellow/magenta tinged skin and black hair and eye brows. He could have

had cyan skin and yellow hair if I had wished to editorialize.

I think several cartridges including Video Byte and Super Snapshot 5 have this false color printing capability. I cannot however with my FINAL CARTRIDGE III control the intensity of exposure and therefore fine tune which features have which color.

I also have experimented with printing from my FINAL CARTRIDGE III each of the low, medium and high exposure files. Each is first converted to a doodle so it can be seen, and frozen, on the screen. With this approach, I can use a single multicolored ribbon on my Star 1000 Rainbow or my color cartridge in my Epson ink jet. The FINAL CARTRIDGE III will print only the colors I have selected after freezing. I chose all red for the foreground for normal exposure, all cyan for low exposure, and all yellow for high exposure.

There is no color scanner that I know of for our Commodores. False color does allow us to reproduce our black and white images in the colors of our choice. The frozen screen approach of FINAL CARTRIDGE III is the most automatic, but allows the least control over colors. The 4-scan method requires that the original program be purchased, but then allow us to act as developers in a darkroom.

The more artistically talented among us, who have good mouse/eye coordination may wish to import the monochrome image to a paint program, and color by hand/mouse/joystick, as Hugh McMenamin has.

Hugh McMenamin's Illustrator for TWS, in its different output versions (for Print Shop and Runpaint or for FGM), is available from CMD and LOADSTAR (see our ad on page 7).

Ken Robinson wrote a most excellent color printer, called Rainbow Printer, available on LOADSTAR #149 as a back issue. If you want to try your hand at color printing, you should check it out.

The Internet For Commodore Users Released

by Gaelyne R. Gasson. The Internet is a big place and it's not always easy to find all the ins and outs about taking advantage of its features when checking it out for the first time. There's a lot of

Commodore specific information available on the Internet, but you have to know how to get to the information, and that can be a Catch-22.

How do I get on the Internet? What term programs can I use? What the heck is FTP? Can I browse the World Wide Web with my C64?

Gaelyne Gasson (formerly Gaelyne Moranec), former editor of CEE-64 Alive!, has been writing about Commodore computers and modeming for several years, with columns and features in BBS Magazine and Commodore World. She has also written about Commodore computing for other publications such as: C=Hacking, Commodore Network, and LOADSTAR. After answering countless letters helping others learn the ropes of telecommunications, she's written a book specifically for Commodore users who want to take advantage of the Internet.

The Internet For Commodore Users is to be packaged with a special version of NovaTerm v9.6 Lite by Nick Rossi. The Internet for Commodore Users covers the following subjects:

- Hardware Basics
- Terminal and Modem Basics
- Finding an Internet Provider
- Getting Online and Signing Up
- UNIX Shell Account Basics
- Welcome To The Internet
- E-mail
- Text Editors
- Newsgroups
- Telnet, Rlogin and Other Utilities
- FTP and Archie
- World Wide Web
- Gopher
- Internet Relay Chat (IRC)
Advanced E-mail Topics:
- Offline Mail Reading
- Sending and Receiving Files
- Mail Lists
- Using E-mail to access FTP, WWW and Gopher
Changing Your Online Environment
- Dealing With Files
- A comprehensive Glossary
- Commodore Term Program Key Equivalents
- Products Source List
- Internet Resource List

The Internet for Commodore C64/128 Users has a PVC coated cover for durability and a plastic comb binding. The book is being published by VideoCam Services and should be ready for worldwide distribution by about the 11th of April.

VideoCam Services is a registered Australian business (0323082E) owned by Rod and Gaelyne Gasson.
Ordering information: (All prices given below are subject to change) The Internet for Commodore C64/128 Users ISBN: 0-646-31615-X $36.95 Australian plus shipping. If ordering from Australia, the cost is $36.95 plus $5 shipping to anywhere in Australia.

If ordering from USA and paying by check, please make the check payable to "VideoCam Services" in the amount of $40.95US. ($29.95 plus $11.00 shipping).

Money orders of any kind can not be accepted. Credit Card Orders:

VideoCam Services can accept most major credit cards including: MasterCard, Visa, Bankcard and American Express.

If ordering from outside Australia (USA, Canada, Germany etc.) and paying by credit card, your card will be billed in *Australian dollars* $36.95 plus $15.00 shipping. This is approximately $40.52US, but varies depending on the rate of exchange. For advance orders, your card will not be debited until The Internet For Commodore Users is dispatched.

Orders can be accepted through postal mail, E-mail, or from the World Wide Web at http://hal9000.net.au/~moranec/bookord.html. Phone orders will also be accepted.

The prices above are for single copy orders and subject to change. Please contact VideoCam Services for bulk pricing.
VideoCam Services
90 Hilliers Rd.
Reynella, SA 5161
Australia
+61 8 8322-2716
gaelyne@hal9000.net.au
http://hal9000.net.au/~moranec
* QWKRR: http://hal9000.net.au/~moranec/qtoc.html*

Graphic Hardware for your 64

by Scott Eggleston. Let's face it, some of us aren't the artists we wish we were. All that experience I had doodling in high school has never translated to an art career. In fact, the doodles I've attempted on my computer have looked pretty awful. Whether I try to use a mouse or a joystick, it rarely looks good.

So what's a person such as myself to do? Importing custom graphics can be very rewarding, giving your project a special flair and uniqueness. While most

of us import such things to spice up a publication, you may also want to bring a graphic touch to a program. All things graphic are possible, and most are easier when using devices that bring existing pictures into our machines.

Of course, I'm referring to digitizing hardware, which simply translates light into pixels that are "on" and dark into those that are "off". The end result is (hopefully) a graphic representation of whatever image it was we wanted to import.

ComputerEyes: The first device of this type I owned was a ComputerEyes module, built by Digital Vision. This black box plugged into the user port and had two large knobs built into its side. One was used for calibrating the gizmo via software, while the other adjusted contrast. A small RCA jack protruded from the opposite side of the box and was your input for any video source.

After an easy setup and calibration, I plugged in a video camera and began shooting. With various levels of "grabbing" you could get a quick, high-contrast picture, or wait awhile and get a "multi-pass" shot with more detail via dithering. All pictures were rendered in the 64's hi-res mode.

One thing was evident: the wait. With the 64's slower processing speed, you were waiting anywhere from 30 seconds to a couple of minutes depending on your capture setting. As a result you could only really shoot something that would not move, i.e. your classic "still life" shot.

The best way, I found, to get a shot of a human or animal subject was twofold. You could either point the camera at a photo of someone, or record the image you wanted on videotape, then pause the playback, and then grab it.

Another bonus with the ComputerEyes was the software. Not only was it well written, but add-on discs existed letting you save your pictures in any of the popular hi-res formats. These included Print Shop, Doodle!, Newsroom, and several others.

Today, you would not need these specialty disks. Using a capture cartridge such as Super Snapshot, it would be easy to grab an image, display it, interrupt the program, and save the image in one of several formats. Like Jeff, I'm of the opinion that everyone needs one of these great little carts. They are just so handy! Anyway, back to the article...

The documentation was also above average, and very informing. The instructions went beyond basic

operations, and included info about how to incorporate pre-fab ML routines in BASIC which would run the unit. An example in the manual suggested running a security camera, which would be constantly scanned. When differences in the scan would arise (via an intruder), the program would report it. Neat stuff, even if no one ever really took advantage of it.

In the end, I had mixed results with the ComputerEyes. Details were easily lost on wider shots (such as landscapes), but improved on close-ups. Light also played a big part in the outcome. Bright lights which washed an image out, would appear even more so with the digitizer. You really needed even lighting and a close camera for a decent result.

I sold my unit a few years back, and have since wished I didn't. Despite its flaws (which were more based on the 64), it was pretty neat, with documentation that was more than expected.

While the ComputerEyes is no longer available new, CMD sells a similar product dubbed "Video Digitizer" for $189.00. Personally, I have no experience with this item, but if a digitizer is the way you want to go, you don't have many alternatives.

The HandyScanner: Another cool gadget that will let you slap real-world images into your Commodore is the HandyScanner. The only scanner ever made for us 8-bitters, the HandyScanner works well enough to give us what we want.

The HandyScanner looks like any other hand scanner you may have seen with one cool twist--it plugs into your Commodore! Actually it plugs into an interface which plugs into the user port. A power supply also plugs into the interface, drawing current from the wall instead of your computer.

The HandyScanner also comes with fairly easy to use software, albeit the manual (a translation from German) is pretty bad. If you're used to GEOS, then you'll like the "point-and-click" style of software used to drive the scanner. Not only will it scan stuff, but is a pretty decent art program as well.

Scanning is an easy process, once you know how to do it. To scan, activate the scanner via software, press the button, and drag the HandyScanner over the image. After about 20 seconds--presto!-- The high resolution image will appear. The software also allows you to enlarge or reduce the image during the scan, which is a great help in achieving certain effects.

What the HandyScanner does best is

scan line art. This includes your doodles, comic art, or anything that is drawn. For color images, the contrast adjustment on the side of the scanner will remove the color (which tends to interfere), and give you a cleaner result.

Scanning pictures is possible, but more difficult. As with the ComputerEyes, the rule is close-up and evenly lit. Small shadows become prominent, and soft lights wash out detail. As with anything, experiment, experiment, experiment!

Although the included software is nice, I prefer geoPaint as an editing tool. Converting the images is easy with HandyImport, a freeware program by Joe Buckley. This gem will quickly take a HandyScanner document and make it a geoPaint. It's fast, easy, and effective. Once in geoPaint, there is a slew of things you can do to your image with any of the many tools available. HandyImport should be included with the scanner when purchased.

The HandyScanner is still available from CMD for $249.00. It may seem pricey, but I've used mine often, and have never regretted purchasing one. They are a must-have for anyone producing a publication.

GeoGIF: Randy Weems' geoGIF opens up a whole world of graphics to your machine. The GIF (Graphics Interchange Format) standard was created by CompuServe waybackwhen, and is still used by many today. The Internet has a slew of pictures in this format, easily converted to geoPaint with this program. Just download the image, boot GEOS and geoGIF it!

This program will also allow you to take images grabbed on other platforms and plop them into your Commodore. Know someone with a flatbed scanner? Have them scan some pictures for you, and save them as a GIF. I have a video framegrabber (ironically, it's a ComputerEyes) on my wife's 286 which allows me to snag live images, which I save as a GIF.

Next, you'll have to get your newly-created GIFs onto your Commodore. If your GIFs were scanned/grabbed on a PC, then you can use a program such as Big Blue Reader or Little Red Reader to get them on a Commodore-formatted disk. Any other platform will require some creativity on your part (such as a null-modem transfer), or have them send the file to your e-mail address. It will be easy to download them from your Internet Service Provider.

If you prefer to have your graphic as a Doodle! file, then you need geoViewer.

this goodie will let you take any portion of a geoPaint file and save it as a Doodle! This program even runs outside of GEOS, but you'll still need GEOS to run geoGIF. GeoViewer has been published several times on LOADSTAR.

Alternative Methods: I know that some of the techniques mentioned above are not cheap, and are far out of reach of some users with shallow pockets. There are, fortunately, other ways to make custom graphics inexpensively, albeit with less desirable results.

All of these methods involved drawing the image yourself. One way is via a graphics tablet such as a Koala Pad or Animation Station. These plug into a joystick port and work with an art program that can drive them (GEOS and OCP Art Studio can handle these). Simply place an image you want to reproduce on the tablet, and trace over it with the stylus. Once a rough sketch is produced, it will be up to you to clean it up with the tools the program provides.

I remember another technique from an old issue of RUN magazine. This involved tracing an image on clear acetate, and then placing the acetate on your monitor. Static electricity holds the clear plastic in place while you reproduce the lines "under" the image with your art program. Once traced, you can alter the image at will. I have never tried this myself, and while the results will never be scanner quality, it is very affordable.

My final suggestion is to literally draw on the screen with a light pen, if you can find one. Software that comes with the pen will support it, or you can use GEOS (64 version only), which will also allow you use of this piece of hardware via an included driver. Of course, this method requires you to have more talent than any of the others previously mentioned.

Many ways exist to help us with our lack of artistry. People with talent can benefit from these avenues as well, but it's nice to know that some of us with more heart than ability can make some cool stuff too.

An Internet Toolkit

By Scott Eggleston. To me, the Internet is just plain groovy. Although we are limited to accessing the 'net in a text-only form, it is still the greatest information resource for us to tap into.

Most of us are aware of the software used to access this phenomenon, i.e. the terminal program. NovaTerm is probably the most popular, as it runs in the most-

used 64 mode. Dialogue and Desterm seem to be the 128er's programs of choice, while a smaller minority are probably using ACEterm in either mode.

Getting the most out of the 'net, does not just mean using your term to its fullest. Other programs exist, and should be taken advantage of, to utilize all that the Internet has to offer.

A Good Text Editor: There are tons and tons of text in Cyberspace. FAQ (Frequently Asked Questions) files for every subject abound. Historical text from the Federalist Papers to the Bible are complete, and word for word. Film Scripts, technical manuals, and documentation are just a few of the blobs of text you can grab for future use.

Once downloaded, how will you view (and/or edit) it, especially if it's a huge file? No word processor available to us can handle the large sizes of text you'll find, so you must look to another kind of program.

The best text editor available to us is Craig Bruce's ZED, published on LOADSTAR 128 Quarterly #30. A program for the 128, ZED does everything a WP can do, except spellcheck and word-wrap. This matters little however, since ZED is not meant for composing, but viewing and editing, which it does very well with simple keystrokes found in the documentation.

Where ZED outshines the WP world is with a Ram Expansion Unit, such as the 17xx series from Commodore, or the clone made by Software Support International. ZED will use any REU memory given it, allowing files of large sizes to reside within. I expanded my 1700 REU to 512k mostly so I could use it with ZED, which Craig claims can use up to 8 megs. I've never known of an REU to go past 2 megs, but if it were possible, I know ZED could handle it. It is a well written, fast, efficient program.

Since upgrading, I've never had a text file that ZED could not examine and edit in its entirety. It pages quickly through the text, can flip to the top or bottom instantly, can delete any amount of text between user-defined marks and resave. It's a great piece of software, unrivalled by anything out there--and it's free!

Users of the 64 can get a taste of ZED in an incomplete version that runs through Craig's ACE operating system. Once loaded, typing 'z [filename]' will load the file. This version of ZED can emulate 80 columns on the 64, as well as use the REU like the 128 stand-alone version. The major difference is that you

cannot set marks or delete chunks, but it is still excellent as a text viewer. Craig is still developing ACE, and has promised ZED upgrades.

The stand-alone version of ZED for the 128 can be found at the FTP site: ccnga.uwaterloo.ca in the pub/cbm/ utils128 directory. ACE is at the same site in the pub/cbm/os/ace directory. Both are freeware.

If you do not own an REU, you still have access to on-line text editors if you have a Unix Shell account. Pico and emacs are the more common (I prefer the simplicity of Pico) and beyond the scope of this article. Lets just say Pico is very similar to Pine, with which you compose e-mail.

The simplest of all methods of viewing is to just create a bookmark when on the web with Lynx (press 'a' to add it to your file). This way someone else can store the info for you and you can go to it anytime you like. Providing they keep their web site up, that is.

Hard Copy: What if you want to print these files? Again, you could load them into your favorite WP, and edit each page for output, but why would you want to? Firstly, it's a total bore, and secondly, the aforementioned large files may not fit. You could make a printout from ZED, but ZED--great as it is--does not give you page breaks.

Enter Ed Bell and LOADSTAR 128 Quarterly. He wrote a fine program called MultiList found on issue #26. MultiList will take a text file of any size and print it out with a custom header (designed by you), and skip all paper perforations giving you a clean, wonderful result. This is just one of the feats this program will do and is worth checking out. It's a real sleeper. Unfortunately, I am unaware of a similar program for the 64.

A Slew of Sounds: Like text, there is a lot of downloadable graphics and text we can snatch as well. My other article talks more about the graphic end of this, but what about those sound files? Rest easy, if you find sound files in the .wav format (and they are without number) you can download and play them on both your 64 as well as your 128.

For the 64, the program you want is Mark Dickenson's Digi-Player. Once booted, you are given a lot of options (on a colorful menu, I might add) which I won't go into, but are covered in the docs. We want to hear those .wavs!

Move the highlight bar to 'load sample' (you are allowed to check the directory, but no file requester is provided), and press RETURN. You'll flip

to the other side of the menu, where you cursor down and click on 'load Atari ST'. This is supposed to convert an ST format sample, and will also work on .wavs. The screen will indicate "changing", and shortly thereafter click on 'play sample', and the .wav will play nicely on your Commodore.

I found when doing this that two things will help the sample to sound better. One, the speed of the sample comes out too slow. Adjust playback speed from 150 to 37. Two, change the video mode to 'off'. This will increase the clarity of the sample, as the computer is freed from screen updates.

Digi-Player does much more, which I'll leave to you to find out. I'm pretty sure DP is also at the ccnga.uwaterloo.ca FTP site. You may have to browse to find it.

Nate Dannenberg's Sound Studio 3.8 will play wavs on the 128 with point-and-click ease. No converting necessary here, just run and click.

Unlike Digi-Player, SS3.8 is mouse driven, has a file requestor, and will even use a Stereo SID cart, if you have one. Just click on the appropriate icon and a directory of any given disk can be scrolled through. To load a .wav, click on it, then click on 'play'. That's all there is to it.

Sound Studio 3.8 is available from the Arkanix Labs website (www.arkanixlabs.com), and is freeware. SS3.8 and upgrades are supposed to work with their recently released 8-bit sampler. I think I'll have to bug 'em for a review model.

As you can see, taking advantage of the Internet means more than just surfing. Download! Edit! Play! Convert! While more elusive, the perks of the 'net are there for us to use as well as anyone else. So use 'em already, and benefit from fine software by fine authors.

Commodore Gazette Still Going Strong

Christopher Ryan, another teenage publisher a la Paul McAleer, started publishing in October 1995. Almost two years later, he's still here.

He had humble beginnings, but his publication grows as time goes by. It's a monthly disk. (3.5" or 5.25" disk: 1541, 1571, or 1581 - request) Subscriptions: USA: 1yr $22, 2yr $44, 3yr $66, 4yr $88 Canada: 1yr $25, 2yr $45, 3yr $70, 4yr $90 IN USA FUNDS!

More 8-Bit Timothy McVeigh Sightings

by Caped Crusader, CommNet. I just watched a biography of Timothy McVeigh on A&E on TV. He's the one on trial here in Denver for the OKC federal building bombing. Well, what caught my eye was that they said his interest as a teenager was guns and computers. Guess what kind of computers? They showed pictures of his Commodore 64 and 1541 disk drive. He even wrote software on the Commodore that impressed his teacher at school.

He even set up a computer bulletin board as a teenager and I suspect it was on a Commodore from what I saw in the biography. Commodore - The only way to fly - Batcave BBS, Denver

Adventures Programming The SuperCPU

By Jeff Jones. There are a few ways you can get in trouble when writing programs for the masses. The worst way is if your code simply fails to work or plain crashes. But there are degrees of failure. Now that some Commodore C-64 computers are running 22 times slower than yours, you can actually think you're writing a useful, zippy program, not fully realizing that on a stock Commodore certain routines could be prohibitively slow. For instance JPEG decoding takes a lot of computing power. Currently there are no JPEG decoders for the C-64. If there were, I'd imagine it would take a whole two minutes to decode a 640x400 image provided you had a RAMLink with adequate space and a SuperCPU. That translates into 80 minutes without a SuperCPU, and maybe 200 minutes with a HD or 1581 disk.

We don't have to visit the hypothetical to realize that there is now hardware shock in the Commodore community. Recently I wrote a program

called Smart Boot, a program which begins printing one line too low on the screen and thus scrolls one line. I received a letter or two about the matter, but dismissed them because when I tested the program, it didn't scroll.

Or *did* it?

I found out yesterday that the program scrolls every time it's used. I never saw it because it scrolls 20 times faster on my system — much too quickly for me, a mere human, to notice. A small scroll isn't all you could miss. Say you've written a cool database with thousands of DIMensioned strings. Your program is riddled with annoying garbage collections of five or ten seconds, but you never notice because these garbage collections only last a quarter second to a half second on *your* machine. If you're writing software for use on other people's computers, you must be conscientious and avoid garbage collections and waits. If you write a complex routine that causes a noticeable pause on your SuperCPU, keep in mind that the same pause will take 20 times longer on a stock C-64. So having a SuperCPU is no license to write sloppy code.

It gets even deeper. You'd think the SuperCPU would be less compatible than a stock C-64, but there are actually programs that work on my SuperCPU that *don't* run on a stock C-64. This tortured me last week when Fender kept reporting a bug in my directory routine. Three times I re-linked the program and three times it gave him a blue screen instead of a directory. It turned out that I had a typo, transposing two digits, in my SYS call to $CE00. So I was SYSing into the middle of the ML. For some reason the SuperCPU forgave the transgression, and worked *every* time while Fender, who fortunately had his SuperCPU switched out, kept getting the crash.

This isn't the only time a forgiving CMD operating system caused us problems. In our re-release of CAD-M, we found out that it's dangerous to use

```
load "filename",peek(186)
```

because without JiffyDOS, this always results in "press play on tape," even if you poke186,8 right before calling. So again, we're finding ourselves forced to test programs without the SuperCPU and without JiffyDOS on a system that exposes sloppy code and, at least to me, seems more strict.

I also have made the mistake of leaving JiffyDOS commands in my programs. I like using JiffyDOS

commands in my program because they are easy, they work as expected, and they are *available*. It's a good idea to make sure you replace JiffyDOS commands (commands Microsoft *should* have placed in BASIC anyway) with the digital sleight of hand we have to pull in order to do simple things like bload a file.

Say, do I sound spoiled? Maybe I should. Once you've tried a Commodore 64 at 20 MHz, you hate to get out of the 20 MHz mode. Lately I've realized that I develop software on a 20 MHz C-64 with RAMLink attached and parallel hard drive cable. You can't get a faster Commodore system — at least not this year. It's probably the hardest advice to follow, but every programmer should also try out new software on a naked C-64 — just to make sure your software is bearable for the masses.

Knees Calhoon Doodles Coming to a Postcard Near You

Knees Calhoon's doodles are legendary at Softdisk. Still it strikes me that most of his art looks infectious — and I mean that in a *bad* way. If I saw any of this stuff under a microscope, I'd scream *"Ebola!"*

His striking but sometimes disturbing art will appear on postcards to be sold privately by Fender Tucker. I would never let a man who drew these images, under the influence or *especially* sober, date my daughter. Would you?
[KC's Comment:] Sorry. I'm not dating anyone at this time.

Nevertheless I thought I'd take some of his more striking doodles and present them.

Knees Calhoon Very Weird Gallery
Images Doodled During Softdisk Meetings
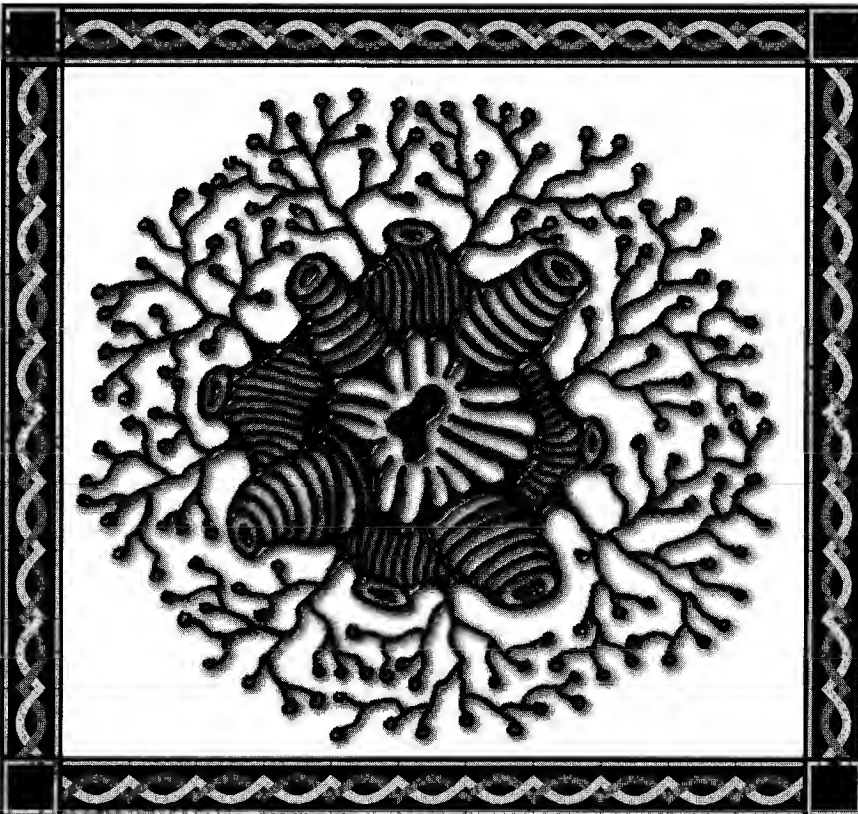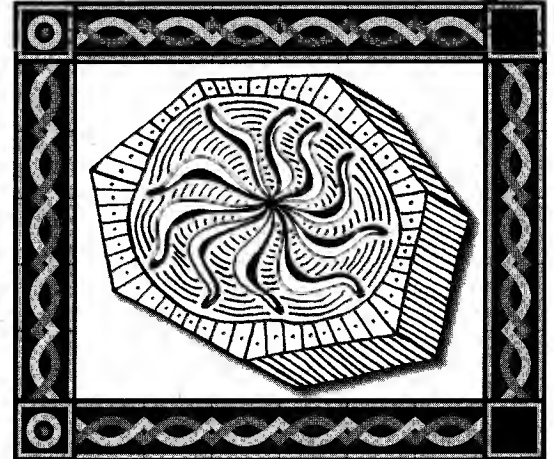
Internet Self-Help: How to Argue and Win Every Time

From Evan Hart's joke list. I argue very well. Ask any of my remaining friends. I can win an argument on any topic, against any opponent. People know this and steer clear of me at parties. Often, as a sign of their great respect, they don't even invite me. You too can win arguments. Simply follow these rules:

Drink liquor: Suppose you are at a party and some hotshot intellectual is expounding on the economy of Peru, a subject you know nothing about. If you're drinking some health-fanatic drink like grapefruit juice, you'll hang back, afraid to display your ignorance, while the hotshot enthralls your date. But if you drink several large martinis, you'll discover you have STRONG VIEWS about the Peruvian economy. You'll be a WEALTH of information. You'll argue forcefully, offering searing insights and possibly upsetting furniture. People will be impressed. Some may leave the room.

Make things up: Suppose, in the Peruvian economy argument, you are trying to prove that Peruvians are underpaid, a position you base solely on the fact that *you* are underpaid, and you'll be damned if you're going to let a bunch of Peruvians be better off. DON'T say: "I think Peruvians are underpaid." Say instead: "The average Peruvian's salary in 1981 dollars adjusted for the revised tax base is $1,452.81 per annum, which is $836.07 before the mean gross poverty level."

NOTE: Always make up exact figures. If an opponent asks you where you got your information, make THAT up too. Say: "This information comes from Dr. Hovel T. Moon's study for the Buford Commission published on May 9, 1982. Didn't you read it?" Say this in the same tone of voice you would use to say, "You left your soiled underwear in my bathroom."

Use meaningless but weighty-sounding words and phrases. Memorize this list:
- Let me put it this way.
- In terms of
- Vis-a-vis
- Per se
- As it were
- Qua
- So to speak

You should also **memorize some Latin abbreviations** such as "Q.E.D.", "e.g.", and "i.e." These are all short for "I speak Latin, and you don't."

Here's how to use these words and phrases. Suppose you want to say, "Peruvians would like to order appetizers more often, but they don't have enough money."

You never win arguments talking like that. But you WILL win if you say, "Let me put it this way. In terms of appetizers vis-a-vis Peruvians qua Peruvians, they would like to order them more often, so to speak, but they do not have enough money per se, as it were. Q.E.D." Only a fool would challenge that statement.

Use snappy and irrelevant comebacks. You need an arsenal of all-purpose irrelevant phrases to fire back at your opponents when they make valid points. The best are:
- You're begging the question.
- You're being defensive.
- Don't compare apples to oranges.
- What are your parameters?

This last one is especially valuable. Nobody (other than engineers and policy wonks) has the vaguest idea what "parameters" means.
- Here's how to use your comebacks:
- You say: As Abraham Lincoln said in 1873...
- Your opponent says: Lincoln died in 1865.
- You say: You're begging the question.
- You say: Liberians, like most Asians...
- Your opponent says: Liberia is in Africa.
- You say: You're being defensive.
- *Compare your opponent to Adolph Hitler.

This is your heavy artillery, for when your opponent is obviously right and you are spectacularly wrong. Bring Hitler up subtly. Say, "That sounds suspiciously like something Adolph Hitler might say," or "You certainly do remind me of Adolph Hitler."

So that's it. You now know how to out-argue anybody. Do not try to pull any of this on people who generally carry weapons.

QUOTE OF THE WEEK:
"Don't force it, get a larger hammer."
-From E-Funnies
Please read the disclaimer:
http://studentweb.tulane.edu/~ehart/ wjdisclaimer.html Wild Jokers Humor List To subscribe, send a message by Evan Hart with SUBSCRIBE in the subject ehart@mailhost.tcs.tulane.edu field. To unsubscribe, do http:// studentweb.tulane.edu/~ehart/ the same but put UNSUBSCRIBE New Orleans, Louisiana, USA as the subject.

LOADSTAR LETTER #44

606 Common Street, Shreveport LA 71101
- **Commodore News**
- **Commodore Views**
- **All Commodore**

Bulk Rate
U.S Postage PAID
Shreveport LA
PERMIT #85